

Digital delay line clock shapers and multipliers

by R. A. Bechade
R. M. Houle

Two digital techniques have been developed to generate an internal clock signal from an external reference clock supplied to a microprocessor. The first method constitutes a clock shaper circuit that produces an output clock that has a 50% duty cycle regardless of the duty cycle of the input reference clock. The second technique generates an internal clock that is an $N/2$ multiple of the frequency of the input clock, where N is an integer greater than 1. Both methods are entirely digital and are independent of process and temperature variations. Their accuracy limits are determined by the technology. Both circuits are described and their results compared.

Introduction

The internal speed of today's microprocessors typically exceeds the bus speed of the systems in which they operate. Microprocessors now include on-chip circuits that produce an internal clock signal whose frequency is a multiple of the system clock frequency. This allows the microprocessor to operate at greater internal speeds while maintaining a synchronous interface with the system. This is especially true for microprocessors which have on-chip internal caches that allow the processor to operate for many cycles without referencing external system memory at the slower system clock rate. Most commonly, such

clock multiplying circuits are analog phase-locked loops (PLLs), which are difficult to design and may require external components or special resistor mask processing layers. There are also many digital phase-locked loops (DPLLs), but most of these still use voltage-controlled oscillators (VCOs) or some analog components [1-3]. Recently, delay-line-loop (DLL)-based clock multipliers [4-12] have been developed.

In this paper we describe two digital techniques which have been developed to generate an internal clock signal from an external reference clock supplied to a microprocessor. The first method constitutes a clock shaper circuit that produces an output clock that has a 50% duty cycle regardless of the duty cycle of the input reference clock. The second technique generates an internal clock whose frequency is an $N/2$ multiple of the frequency of the input clock, where N is an integer greater than 1. Both methods are entirely digital and are independent of process and temperature variations. No external components are necessary, other than a clock input from a crystal oscillator. The accuracy limits of the clock shaper/multiplier reported here are limited by the smallest delay that can be generated on the chip and, to a small extent, by voltage noise on the power supply.

Both designs use variable-length delay lines with internal connections to generate the intermediate delays. One design determines the total number of delay stages equivalent to the input frequency in one cycle. The other method incrementally increases the length of the delay line

©Copyright 1995 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/95/\$3.00 © 1995 IBM

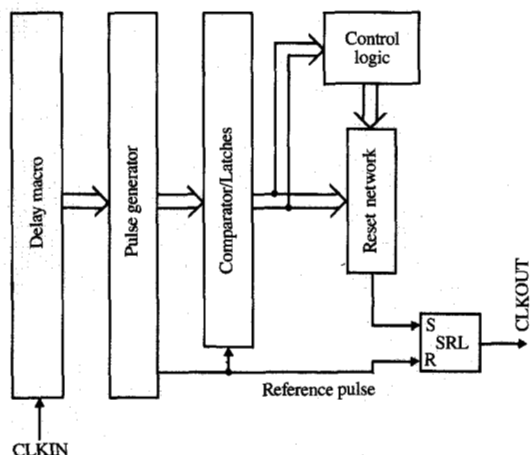


Figure 1

Clock shaper block diagram.

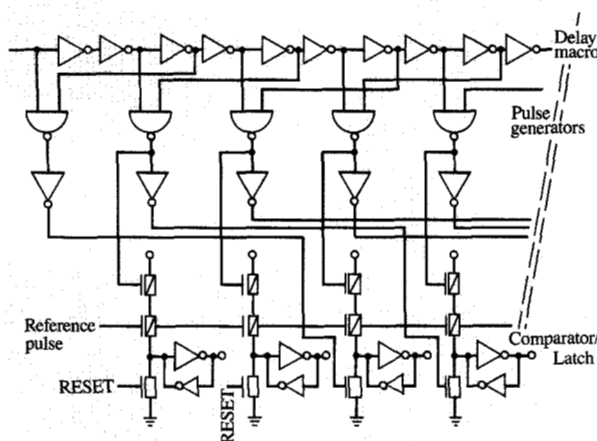


Figure 2

Circuit schematics for the clock shaper.

until its delay is equivalent to the input frequency. In both cases, jitter is a function of the speed of one of the delay elements.

1X clock shaper

The performance of chips using the 0.8- μm photolithographic technologies requires chip clock rates

which are difficult to distribute at the card level. The incoming clock rate is commonly halved to provide an internal signal with a 50% duty cycle. The 1X clock shaper allows the chip to run at the external clock frequency, independent of the external clock duty cycle. This clock shaper does not actually multiply the incoming clock frequency, but shapes it to provide a square wave with a fixed 50% period. It is commonly referred to as a clock multiplier because it allows the chip to run internally at twice the external bus speed. However, internal clock variation caused by process and operating conditions can be significant. This variation affects the system performance through a wide range of specifications for I/O signals. To reduce the spread of I/O timing, the clock shaper circuit is designed to have zero latency between the system clock and the internal clock of the processor. The design objective for the clock shaper was a minimum number of initialization cycles and the ability to stop and restart the external clock at the same frequency without reinitialization. The clock shaper also had to be functional over a wide range of frequencies.

• Circuit description

The basic idea is to propagate the reference clock signal through a delay line and to monitor the stage N of the line that has been reached by the clock signal when the next input transition occurs. A set/reset latch (SRL) is set on the rising edge of the incoming clock signal, and when the pulse reaches stage $N/2$, the SRL latch is reset. This generates a square wave with equal periods of the same frequency as the incoming clock. The same approach can be used to double the frequency by setting the latch both on the rising edge of the incoming clock signal and when the pulse reaches stage $N/2$, and resetting the latch at stage $N/4$ and stage $3N/4$.

Figure 1 shows a block diagram of the clock shaper. The delay macro is an open chain of inverters. Each delay stage contains two inverters. The pulse generator macro produces a narrow pulse at the rising edge of a transition for each of its associated delay stages. The comparator/latch macro compares a reference pulse generated on the rising edge of the incoming clock to the output of all other pulse generators. At the stage where a match occurs, a latch is set. To guarantee functionality, a minimum of three adjacent latches are set. This also eliminates metastability concerns for the comparator latches. To ensure continuous monitoring, the comparator/latches are re-evaluated every other cycle. As the clock signal propagates along the delay line, the complements of the pulses reset the comparator latches several stages ahead in the line. Figure 2 is a simplified schematic of the delay macro, pulse generator, and comparator/latch circuits.

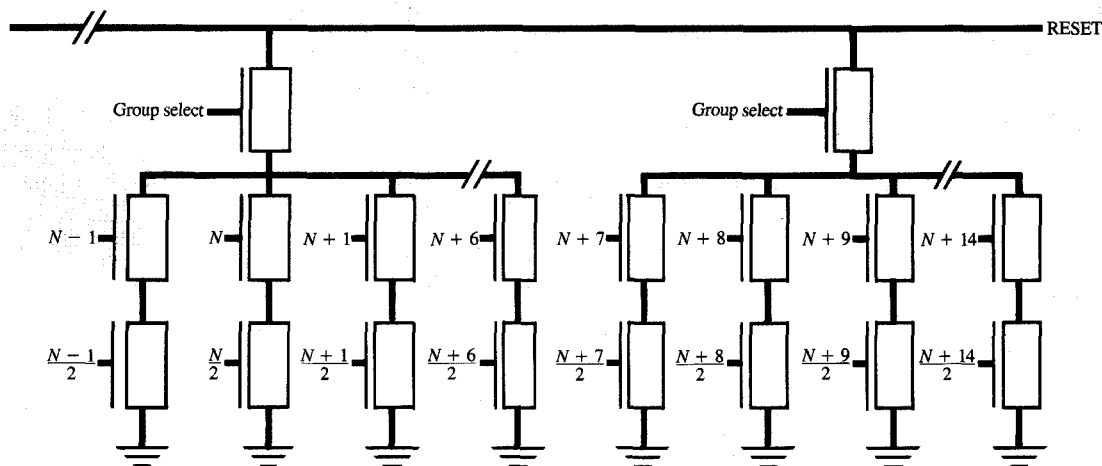


Figure 3

Reset network for the set/reset latch.

The output of the clock shaper is controlled by the SRL. The set input of the SRL is connected to a pulse generator that generates a pulse on the rising edge of the clock. This reference pulse is wider than the pulse generated by the pulse generator macro along the delay line. The SRL is reset at the 50% point of the incoming clock by a reset network, as shown in **Figure 3**. The comparator macro determines the number of stages of delay, N , necessary to delay the input clock by one period. Each comparator/latch is connected to the gate of a pull-down device in the reset network; a second pull-down device has its gate wired to a pulse generator corresponding to an $N/2$ delay to gate the discharge of the reset network. When the next pulse has propagated through $N/2$ stages, the reset network is discharged and changes the state of the output SRL. The point along the delay line at which the pulse repeats varies with the process, temperature, and voltage, but the 50% point is predetermined for every comparator/latch. For improved resolution when N is an odd number, the pulse generated at stage $(N - 1)/2$ is delayed by half the delay propagation of the delay line. To limit the capacitance on its output, the reset network is segmented in groups of eight pairs of pull-down devices by an additional device in series between the common node and the output. This additional device also prevents multiple selection in the reset network. To accommodate all variations in process, temperature, and voltage, the delay line must be quite long (108 stages in this design). With a combination of slow process, low voltage, and high

temperature near worst case, the clock signal may travel through only 30 or 40 stages for each cycle. In this situation, several combinations of comparators/latches will be enabled. The control logic macro (**Figure 1**) does priority encoding on the comparator/latch macro and enables only the first group of the reset network that has a match. When the match occurs at the boundary between two adjacent groups, that situation is recognized and both groups are enabled.

For proper operation, the delay line must be long enough for a match to occur, but for low-frequency operation this is not possible because of the large number of delay stages required. For low-frequency operation, the pulse shaper is wired to reset the SRL when the incoming clock reaches half the length of the delay line. The output pulse is asymmetrical, with a fixed time at the high level and an extended time at the down level; however, at low frequency the chip does not need a symmetrical clock.

One concern with this design is the delay between the external clock and the output of the clock shaper, which may cause some problems for the chip I/O timing. To eliminate this concern, the design of **Figure 1** is modified with a latency eliminator circuit.

• Latency eliminator

The objective of the latency eliminator circuit is to delay the incoming clock by some variable amount so that the internal clock pulse occurs exactly one cycle later than the clock input. The latency eliminator circuit contains a chain

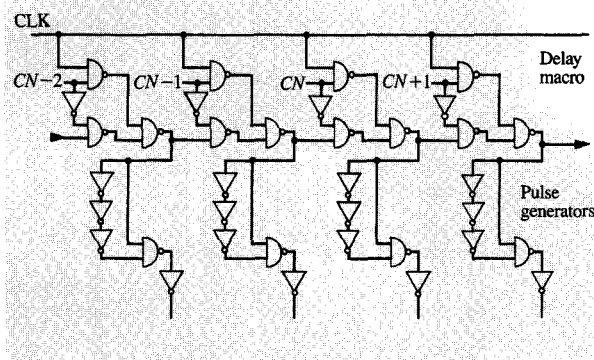


Figure 4

Delay macro and pulse generator.

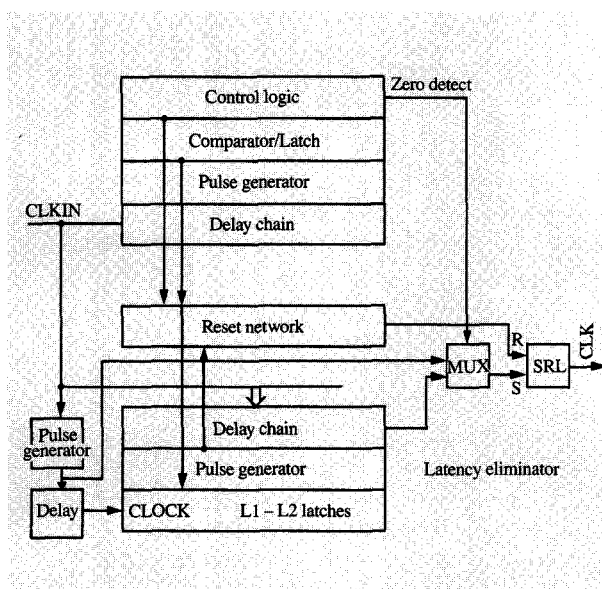


Figure 5

Clock shaper with latency eliminator.

of gated NAND gates with pulse generators, as illustrated in Figure 4. In this design, a second latch is needed in the clock shaper to hold the compared data one extra cycle. The delay macro (Figure 4) is different from the clock shaper because of the need to insert the incoming clock signal at any stage along the delay line. A unique control line for each stage is connected to the output of the second latch in the comparator/latch macro. The stage delay in

this NAND chain is designed to be the same as the inverter chain in the clock shaper section.

When N has been determined in the clock shaper, it selects the point where the incoming clock signal must be inserted in the latency eliminator delay line with the CN control line. The complement of CN turns off the other NAND gate and prevents other signals from propagating. The point at which the incoming clock signal is inserted is hard-wired in the design. Each latch in the clock shaper section of the design is associated with a different stage in the latency delay line. The offset in delay is determined by circuit simulation of the SRL latch and the associated circuitry that contributes to the internal delay of the clock shaper. The sum of the delay through the latency eliminator delay line and the clock generation delay equals one full cycle. It is also possible to compensate for the delay of the clock buffer network so that there is no delay between the external clock and the clock signal at the latches.

The reset network for the SRL must be modified to maintain the fixed 50% ratio. The $N/2$ gating is generated by the pulse generators associated with the latency compensation delay line. The $N/2$ delta is measured from the end of the delay line. When a match is found, a latch is set in the clock shaper. Some time after the rising edge of the incoming clock signal, the data from the clock shaper section are transferred into a second set of latches, and the next clock input is inserted in the latency delay line. When the clock signal in the latency delay line reaches $N/2$ stages before the end of the line, the reset network is discharged, resetting the SRL. When the delayed clock signal reaches the end of the line, a pulse is generated and the SRL is set.

When the input frequency is too low to find a match in the clock shaper, the latency eliminator is bypassed and the internal clock is delayed with respect to the external clock; again, this is not a problem, because at low frequency this timing is not critical. The reset network is then gated from the pulse generators in the middle of the clock shaper delay line. Figure 5 shows a block diagram for the combination of the clock shaper and the latency eliminator.

• LSSD test

A considerable effort has been made to make the design testable by the level-sensitive scan design (LSSD) method. All latches are full L1-L2 and scannable. A 99.2% testability has been achieved with automatic test patterns. Some untestable nodes remain because of the built-in redundancy, with at least three adjacent latches set simultaneously. By design, only two need be set to guarantee proper operation, so there is some built-in margin. The LSSD overhead amounts to 31% of the macro area.

• Experimental results

The design has been fabricated in the 0.8- μm technology and used in the IBM 486SLC2 chip. The total area occupied by the 50-MHz clock shaper with no latency is 0.8 mm², with 108-stage delay lines. The test results show no variation in pulse width from 2.8 V at -10°C to 5.5 V at 70°C . The waveforms are shown in Figure 6. The design point of the circuit is 50 MHz, and it operates at frequencies ranging from less than 1 MHz to 100 MHz. The jitter is at least the delay of one basic delay element, which is 0.4 ns at nominal conditions. The total jitter is a function of the power supply. The design is generally immune to power supply noise of up to 10 ns, but voltage variation over several cycles affects the jitter. We found that we needed to decouple the supply to the clock multiplier from the rest of the chip power supply. Under these conditions, the jitter has not appeared to be a problem, so no detailed characterization was pursued because of time limitations. From a limited sample, the variation on the 50% point is $\pm 1.5\%$ at 50 MHz. The latency compensation scheme is also functional; because of timing considerations, the internal clock signal was selected to occur 0.5 ns before the system clock signal. The same sample shows a variation of ± 0.6 ns. The design is initialized in four cycles and is entirely static. The external clock can be stopped and restarted, at the same frequency only, with no impact on the internal clock. The clock shaper has been shown to be independent of the shape of the incoming clock signal.

N/2 digital clock multiplier circuit

We now describe another digital technique for producing an internal clock signal whose frequency is a multiple of the system clock frequency. This circuit is similar to that described above in that it employs a digital delay line to estimate the period of the incoming system clock signal, but it does so by incrementally increasing and adjusting the length of the delay line. This circuit is capable of producing a multiplied output signal whose frequency is $N/2$ times the incoming system clock frequency, where N is an integer greater than or equal to 2.

Figure 7 shows the basic block diagram of the circuit. Let the CLKIN signal be a periodic signal (with arbitrary duty factor) of period T . The clock generator circuit consists of N variable-length delay lines. The output of each delay line, or leg, drives a one-shot circuit that produces a short positive pulse whenever the leg goes from low to high. These pulses are ORed together to drive a toggle latch. The output of the toggle latch is the output of the overall clock generator circuit. If the delay of each leg is the same, and if the total delay of all legs equals one external period T , each leg represents T/N delay, and the overall circuit output, XOUT, will uniformly change state

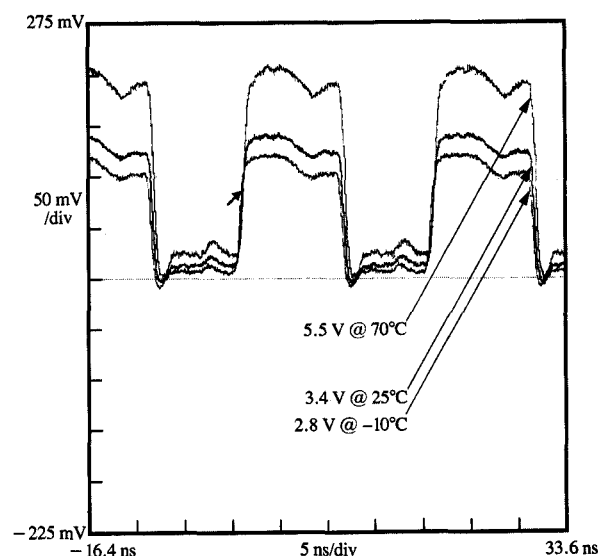


Figure 6

Clock shaper output waveform.

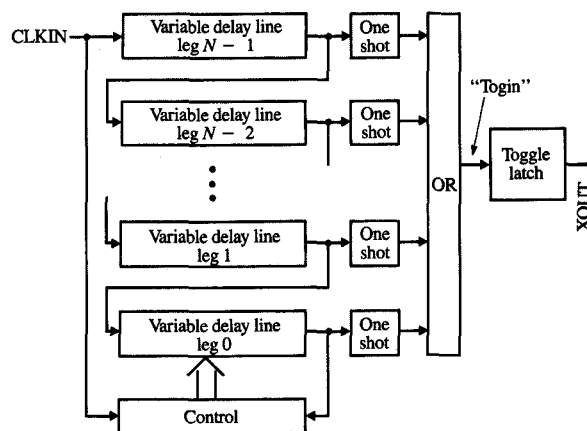


Figure 7

Basic block diagram for $N/2$ clock multiplier.

N times in time period T . Hence, the period of XOUT is $2T/N$, and its frequency is $N/2$ times the frequency of the

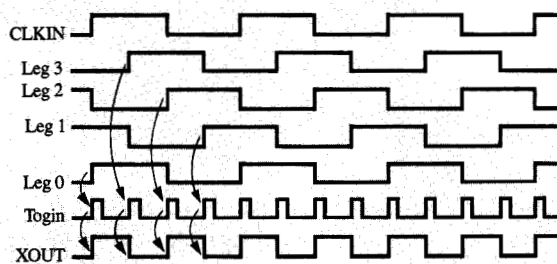


Figure 8

Sample clock multiplier element waveforms for the case where multiplication factor = 2, number of legs $N = 4$.

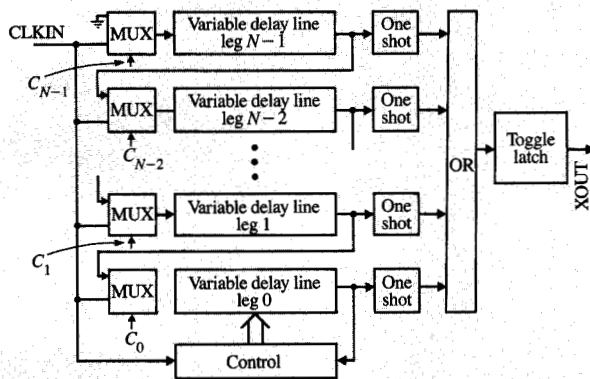


Figure 9

Basic clock multiplier block diagram with input multiplexers to allow different multiplication factors.

incoming CLKIN signal. Figure 8 illustrates the pertinent signals for the case in which $N = 4$ after synchronization has been achieved. (As with the phase-locked loops, this circuit also requires several cycles to acquire synchronization with the incoming clock signal. The time to achieve synchronization varies with input period, process parameters, voltage, and temperature, but typically it is in the hundreds of cycles for an external frequency of 66 MHz.)

Figure 9 shows a minor enhancement to Figure 7. By including multiplexers at the beginning of each leg, one can direct the CLKIN signal to the appropriate delay line so that one of several frequency multiplication values can be achieved. For example, if eight legs were included in the circuit on the die, with the appropriate multiplexing the clock output could be made to be $2/2$, $3/2$, $\dots$, or $8/2$ times the CLKIN signal. Each input multiplexer allows the output of the previous leg to pass if its control signal is low; it allows CLKIN to pass if its control is high. Only one multiplexer control line is high; all others are low. [If its control line is low, the first multiplexer passes a null signal (held at ground), so that all legs that are not needed for the desired multiplication factor are quiescent and do not consume energy.] Thus, in this example, if the desired multiplication factor is 2, four of the eight legs will be active and four will not. The control for the input multiplexer on leg 3 will be high and all others low. Consequently, legs 7-4 are quiescent, and legs 3-0 are active. (There is an OFF signal, not shown in the block diagram, that initializes the clock circuit and inhibits its output. The multiplication factor may be changed while this OFF signal is asserted, but must be stable whenever the clock generator circuit is actually running.)

Figure 10 shows the composition of each leg. The coarse delay block consists of several pairs of inverters which ensure that the delay of each leg is greater than some minimum value so that the positive pulses produced by the one-shots at the end of each leg are guaranteed not to overlap one another. The rest of the leg constitutes the variable portion of the delay line. As can be seen in the diagram, it consists of many pairs of inverters. Each pair of inverters (i.e., buffer) feeds the next buffer and an n-MOS pass gate. The outputs of all of the pass gates are tied to a common node. There are M stages in each delay leg. The total delay from the coarse output, N_c , to the common node, N_{com} , is determined by the pass gate select lines, S1 through S_m . Only one select line is high at any time. If S1 is high, the delay from N_c to N_{com} is the delay of one buffer driving the common node. If S2 is high, the delay from N_c to N_{com} equals the delay associated with two buffers (one lightly loaded and one loaded with the common line). If S_i is high, the delay from N_c to N_{com} is that of i buffers ($i - 1$ lightly loaded and one loaded with the common line). The common line feeds a half-latch buffer that restores the up level of the common node to V_{DD} (since only n-MOS pass gates are used). The final granularity of the delay is controlled by the output multiplexer circuits. The output of each leg is the output of the half-latch, N_{bhl} , or N_{bhl} delayed by one CMOS pass gate (which is designed to be approximately equal to the delay of one inverter), or N_{bhl} delayed by one more lightly loaded inverter pair.

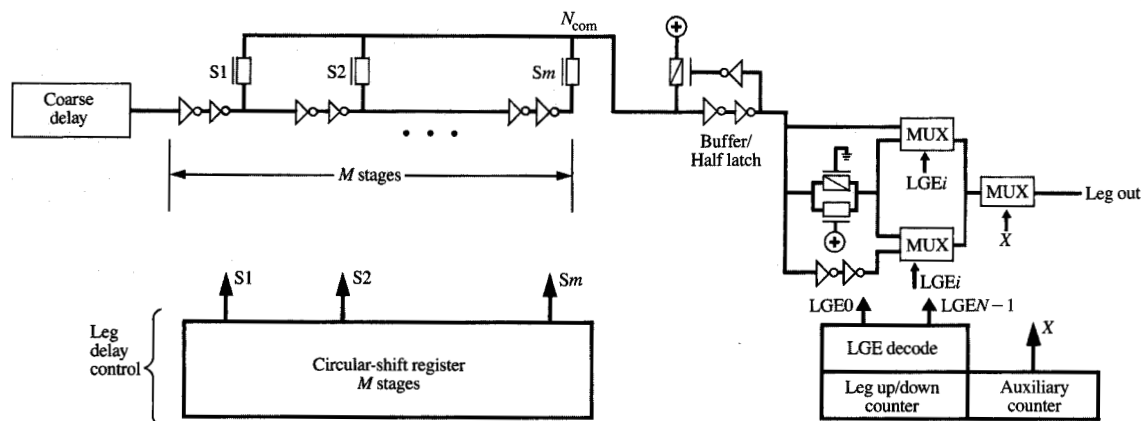


Figure 10

Variable delay line.

To ensure that delay is added to or removed from each leg evenly, the control circuitry consists of a stage shift register, a leg counter, and a one-bit auxiliary counter. The stage shift register determines which select line, S_i , is high, and it affects all legs simultaneously. Thus, it determines the base delay of each delay line. It consists of M master-slave latches arranged in a circular-shift manner so that both right and left shifts are allowed. Initially, while the clock multiplier circuit is off, this stage shift register is set to be 10000...0, so that only select line S_1 is high. When the compare circuitry determines that more delay is required in each leg, it shifts the stage register right one position. When the compare circuitry determines that less delay is required in each leg, it shifts the stage register left one position. Shifting the stage register one position changes the delay of each leg by one lightly loaded inverter pair.

Also shown in Figure 10 are the leg and auxiliary counters, which control each leg's output multiplexor. These counters determine the difference in delay between any two legs. They count in such a manner that the difference between any two legs is never more than the delay of one CMOS pass gate (or approximately the delay of one inverter). The leg counter is a circular counter that ranges from 0 to $N - 1$ (i.e., the number of active legs). The auxiliary counter, X , is a one-bit indicator that changes state whenever the leg counter changes from $N - 1$ to 0 in the forward direction; or from 0 to $N - 1$ in the reverse direction. Simple logic circuits decode the output of the leg counter into the LGE signals that are

unique for each leg. LGE1, which goes only to leg 1, is high whenever the leg counter is greater than or equal to 1. LGE2, which goes only to leg 2, is high whenever the leg counter is greater than or equal to 2, etc. The LGE signals and the auxiliary counter determine how much additional delay each output multiplexor will add to each leg's base delay. If LGE and the auxiliary counter are both low, no additional delay is added. If either one is high, but not both, the delay associated with one CMOS pass gate is added. If both LGE and X are high, the delay of two inverters is added to the delay line. The stage shift register changes one position only when the leg counter and auxiliary counters pass a cycle boundary; i.e., the shift register shifts right one position when the leg counter goes from $N - 1$ to 0, and the auxiliary counter from 1 to 0. It shifts left one position when the leg counter changes from 0 to $N - 1$, and X changes from 0 to 1. In this way, the difference in delay between any two legs is never more than approximately one inverter delay.

Figure 11 depicts an example counting sequence. Suppose the multiplication factor equals $1.5X$, so that there are three active legs numbered 2, 1, and 0. Also suppose that the CLKIN period, coarse delays, input and output multiplexor delays, half-latch delays, etc. are such that the ideal number of lightly loaded inverters is 23, 24, and 24 in legs 2, 1, and 0, respectively. Adding one more inverter to leg 2 causes too much delay, and the compare circuitry reverses the counting direction. The example starts after the stage counter has reached the value of 10 and the leg and X counter are 0. (This is arbitrarily called

Figure 11[illegible]

Figure 12

cycle 1 in Figure 11.) Since there are ten stages per leg (i.e., 20 inverters), leg 0 effectively has 21 inverters because LGE0 is high and its output multiplexor adds one CMOS pass gate to the delay. In the next cycle, the leg counter advances to 1, so that both LGE0 and LGE1 are now high. Now, both legs 0 and 1 effectively have an additional

The total number of legs in the circuit is twice the largest desired multiplication factor. The number of stages in each leg is determined by the frequency range and multiplication factors of interest. Ideally, each leg should contain enough stages that the total delay of all legs is greater than the longest period of the incoming CLKIN signal for all process, voltage, and temperature conditions. Thus, the number of stages in each leg can be determined from the following equation: $N_{\min} \text{ legs} \cdot (D_c + (M - 1) \cdot D_b + D_{bhl} + D_{om}) > T_{\max}$, where $N_{\min}$ is the number of legs associated with the smallest desired multiplication factor, D_c is the coarse delay, D_b is the delay associated with a lightly loaded pair of inverters, D_{bhl} is the delay of the half latch plus the delay of the one inverter pair that must drive the common node, D_{om} is the delay of the output multiplexor, and $T_{\max}$ is the period of the slowest input clock.

The second major function of the control logic is to adjust the overall delay so that the output of the last leg equals the CLKIN signal delayed by one period, T . This circuitry is shown in **Figure 12**. Two basic circuits are used: an AND/latch circuit which produces a signal called CLOSE, and a sampling latch. Initially, when the OFF signal is active, the CLOSE latch is reset to 0 and the stage shift register (shown in Figure 10) is set to 10000...0 so that each leg has its minimum delay value. This ensures that the total delay of all legs is much less

than T . The clock signal of the CLOSE latch, CLSCLK, is the AND of two pulses, PGEXT and PGEST. PGEXT ("pulse-external") is a narrow pulse produced whenever CLKIN goes from low to high. PGEST ("pulse-estimate") is a narrow pulse produced whenever the output of leg 0 goes from low to high. The width of these pulses and the minimum delay of each leg ensures that these pulses do not overlap when the circuit is initially enabled by the lowering of the OFF signal. Since CLSCLK is 0, CLOSE remains low. This forces the ADD DELAY signal to be high. Adding delay causes the leg counter, auxiliary counter, and stage shift register to advance as previously described. Eventually, enough delay will have been added to each leg to cause PGEST to begin to overlap PGEXT. This will then force CLOSE to be high. At this point the sampling latch will start to control the direction of change. The sampling latch is a master-slave D-type latch that captures the value of CLKIND on the rising edge of PGEXT. (The amount of delay between CLKIN and CLKIND equals the delay between leg 0 and PGEST minus the set-up time of the sampling latch.) So, if leg 0 rises before CLKIN rises, the output of the sampling latch will be low and ADD DELAY will be high. If leg 0 rises after CLKIN rises, the output of the sampling latch will be high and REMOVE DELAY will be high. This latter condition causes the leg counter to count down and the shift register to shift left, thereby removing delay. Since the control mechanism causes the data and clock inputs of the sampling latch to align with one another, there can be a metastability problem with this latch. A simple gating circuit, which toggles every external cycle, allows the sampling latch one full cycle to stabilize. Hence, the sampling (comparing) of CLKIN to leg 0 occurs on odd cycles, and the updating of the leg and auxiliary counters occurs on even cycles.

The circuit can be enhanced further to perform some limited amount of latency correction. Latency is the delay between the rising edges of the internal and external clocks. It affects the input and output timings of the microprocessor because all timings are in reference to the rising edge of the external clock and not the internal microprocessor clock. In this circuit, latency is caused by the delay of the one-shots, the OR circuit, and the toggle latch of Figure 7. Figure 13 shows a simple modification to the circuit that removes the latency generated by the above output circuits. A delay block D_x , which approximately equals the delay of the output circuits, has been added to each leg as shown in the diagram. The signals that drive each one-shot and OR output leg occur D_x sooner than they did in Figure 7. Thus, the output, X_{out} , rises at nearly the same time as CLKIN rises. This latency correction technique is possible only if the total length of the delay lines exceeds one external period T for all desired input frequencies and multiplication factors.

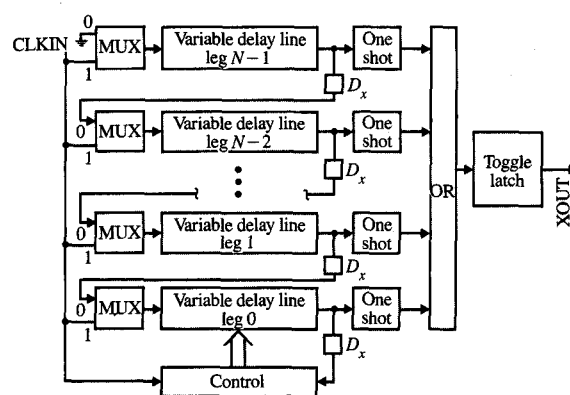


Figure 13

$N/2$ clock multiplier block diagram with latency correction.

This circuit with latency correction has been implemented in an $0.8\text{-}\mu\text{m}$ CMOS technology on the IBM Blue Lightning microprocessor. There it provides a $1X$ ($2/2$) or a $1.5X$ ($3/2$) multiplied internal clock signal for CLKIN frequencies of 40 MHz to 66 MHz. It consists of three legs, each containing 80 stages, and its size is 0.76 mm^2 .

Figure 14 shows the output signal waveform of this circuit. These data were obtained by picoprobng a die on a wafer while the processor was running a small program driven from a tester. The circuit was in $1X$ ($2/2$) mode, so that its output period equaled that of the CLKIN signal. The jitter in the output waveform period is 360 ps peak to peak. Because the tester was looping on a small set of test vectors while these data were taken, the processor itself was probably not being fully exercised, and V_{DD} noise was at a minimum.

A more realistic picture of the output waveform is shown in Figure 15, which was obtained by picoprobng a special Blue Lightning test module in which the die was exposed. Normally, the die is mounted face down in the standard Blue Lightning module. For debugging purposes, some chips were mounted face up and the modules were left open. These special test modules required several sockets to correctly wire them to the planar. The CLKIN frequency was from a 50-MHz crystal oscillator which had a measured input jitter of 400 ps peak to peak. The system was then booted normally and was running a chess program at $1.5X$ ($3/2$) mode. The output signal shows a peak-to-peak pulse jitter of about 1.0 ns and a period jitter of about 1.4 ns. Because of the sockets and open module,

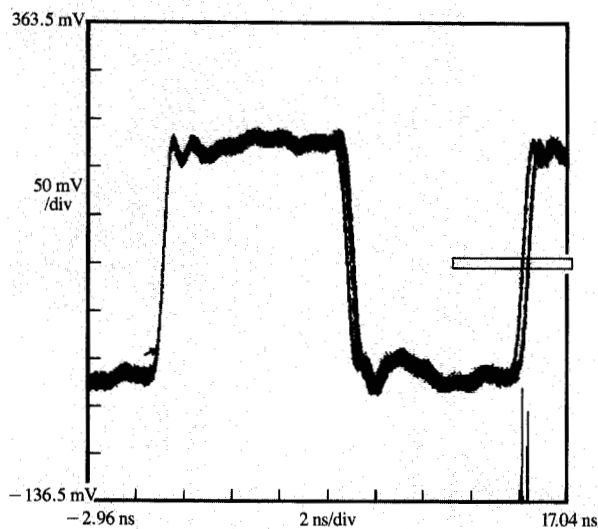


Figure 14

CLKOUT signal on chip on wafer.

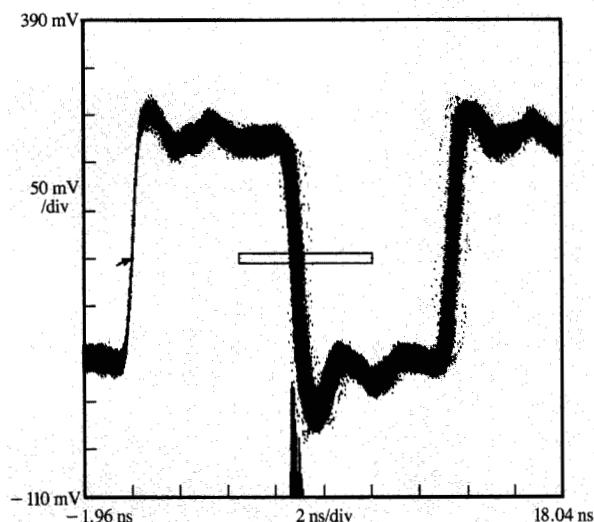


Figure 15

CLKOUT signal in special test module in system.

there was considerable V_{DD} noise in the system. When a separate V_{DD} supply was used to source the on-chip $N/2$ clock multiplier circuitry, the peak-to-peak period jitter was reduced to about 840 ps. Thus, the jitter on the clock multiplied signal was only 440 ps greater than that provided by the input clock. This magnitude of jitter implies that the internal speed of the microprocessor has to be greater than the ideal minimum. For example, an 0.8-ns peak-to-peak period jitter implies that the minimum period produced by the clock when running at 75 MHz is 12.9 ns, rather than the ideal 13.3 ns. Thus, the internal speed of the processor must be 77.5 MHz to compensate for the jitter, rather than the ideal 75 MHz. This additional performance requirement on the processor represents a yield loss which must be compared to the cost of using a more accurate analog PLL with additional pins and external components.

Summary

Two digital delay-line clock shaper/multipliers have been described. The first design, a clock shaper, offers several power-saving features. It needs very few initialization cycles, and the input clock can be stopped and restarted without re-initialization of the shaper. This design could be implemented as a multiplier, but would be limited to double the input clock frequency in practical applications.

The second design offers greater flexibility for the frequency multiplication ratio, since it can be reinitialized with a different number of delay lines. Because it has the capability of multiplying the incoming clock frequency by $N/2$, half-integer multiplication modes are accommodated as easily as integer modes. This can be important in improving the granularity of processor performance sorting. For example, common system clock rates for today's personal computers are 50, 60, and 66 MHz. Having the capability to multiply this incoming frequency by 1.5, 2, and 2.5 allows one to sort a single microprocessor design into many bins to increase the marketability of the overall personal computer. A nominal 90-MHz microprocessor could be used in 50- and 60-MHz systems at 1.5X, greatly increasing overall performance over that of a system which used the same processor without clock multiplication. This same nominal 90-MHz design would yield some 100-MHz parts, and perhaps even some 120-MHz modules. These could be used in 50- or 60-MHz systems at 2X, and 66-MHz systems at 1.5X.

The jitter of both techniques is comparable, although possibly not as good as that which can be achieved with analog PLLs. However, because of their digital nature, they are much easier to design and manufacture than PLLs. Loop gain and stability analyses are not required, nor are the external components or special resistor processing layers so common to PLL designs. In addition, because both of these techniques are not free-running

oscillators, they respond almost instantaneously to the incoming reference clock signal. This can be important in debugging the processor and in power-management systems. One can stop the internal processor clock directly with the external system clock, without disturbing the PLL frequency synchronization mechanism.

Since both techniques described in this paper utilize delay lines composed of repeated elements, the layout and circuit design are easily realized. Also, the latches can be made to be LSSD-compatible so that very high test coverage can be achieved.

Acknowledgments

We thank B. Kauffmann, R. Bishop, D. Pham, and P. Kartschoke for their technical support for the design and simulation of these circuits, K. Shaw for the layout of many of the circuits, and R. Bell, J. Raymond, and G. Rohrbach for their laboratory support.

References

1. W. Lindsey and C. M. Chie, "A Survey of Digital Phase Locked Loops," *Proc. IEEE* **69**, 410-431 (April 1981).
2. D. K. Jeong, G. Borriello, D. A. Hodges, and R. H. Katz, "Design of PLL Based Clock Generation Circuits," *IEEE J. Solid-State Circuits* **SC-22**, No. 2, 255-261 (April 1987).
3. B. Kim, D. N. Helman, and P. R. Gray, "A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2- μ m CMOS," *IEEE J. Solid-State Circuits* **25**, No. 6, 1385-1394 (December 1990).
4. M. Johnson and E. Hudson, "A Variable Delay Line PLL for CPU-Coprocessor Synchronization," *IEEE J. Solid-State Circuits* **23**, No. 5, 1218-1223 (October 1988).
5. A. Waizman, "A Delay Line Loop for Frequency Synthesis of the De-Skewed Clock," *ISSCC Digest of Technical Papers*, pp. 298-299 (1994).
6. T. Rahkonen and J. Kostamovaara, "The Use of Stabilized CMOS Delay Lines for the Digitization of Short Time Intervals," *IEEE J. Solid-State Circuits* **28**, No. 8, 887-894 (August 1993).
7. A. Efendovich, Y. Afek, C. Sella, and Z. Bikowsky, "Multifrequency Zero-Jitter Delay-Locked Loop," *IEEE J. Solid-State Circuits* **29**, No. 1, 67-70 (January 1994).
8. R. Marbot, "Phase-Locked Loop and Resulting Frequency Multiplier," U.S. Patent 5,260,608, November 9, 1993.
9. P. Vanderbilt, "Direct Digital Frequency Multiplier," U.S. Patent 5,257,301, October 26, 1993.
10. F. Sato, T. Saba, D. K. Park, and S. Mori, "Digital Phase Locked Loop with Wide Lock-in Range Using Fractional Divider," *Proceedings of IEEE Pac Rim 1993*, pp. 431-434.
11. W. J. Gleeson and J. R. Young, "Digital Self-Calibrating Delay Line and Frequency Multiplier," International Patent WO 93/13598, July 8, 1993.
12. G. T. Koker and S. T. Tsang, "Apparatus for Generating Multiple Phase Clock Signals and Phase Detector and Recovery Apparatus Therefor," International Patent WO 92/00558, January 9, 1992.

Received June 21, 1994; accepted for publication November 29, 1994

Roland A. Bechade IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (U6931 at BTVLABVM). Mr. Bechade received his B.S. in electronics from the École Supérieure du Canton Vaud, Switzerland, and his M.S. in computer science from the University of Vermont, Burlington. He joined IBM in 1965 and since 1978 has been involved in the design of microprocessors. Mr. Bechade holds nine patents and has earned four IBM Invention Achievement Awards.

Robert M. Houle IBM Microelectronics Division, Burlington facility, Essex Junction, Vermont 05452 (U6725 at BTVLABVM). Mr. Houle received his B.A. in mathematics and B.S. in electrical engineering from the University of Vermont, Burlington, and his M.S. in electrical engineering from the University of Connecticut at Storrs. Mr. Houle joined IBM in 1980 and since 1981 has been involved in the design of microprocessors.